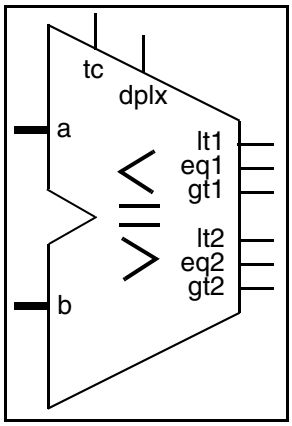
DW_cmp_dx

Duplex Comparator

Version, STAR and Download Information: IP Directory

Features and Benefits

Revision History



- Selectable single full width Compare, or two smaller width Compare operations (duplex)
- Selectable number system (unsigned or two's complement)
- Parameterized full word width
- Parameterized partial word width (allowing for asymmetric partial width operations)
- Separate flags for Less Than, Equal To, and Greater Than
- Two sets of flags for duplex operation

Description

The DW_cmp_dx compares operands `a` and `b` as either:

- A single comparison of *width* bits (simplex mode [`dplx` is 0]), or
- Two comparisons: one of *p1_width* (least significant) bits and one of (*width* – *p1_width*) (most significant) bits (duplex mode [`dplx` is 1]).

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
a	*width* bits	Input	Input data
b	*width* bits	Input	Input data
tc	1 bit	Input	Two's complement control
dplx	1 bit	Input	Duplex mode select (active high)
lt1	1 bit	Output	Part1 : less-than output condition
eq1	1 bit	Output	Part1 : equal output condition
gt1	1 bit	Output	Part1 : greater-than output condition
lt2	1 bit	Output	Full width or part2 : less-than output condition
eq2	1 bit	Output	Full width or part2 : equal output condition
gt2	1 bit	Output	Full width or part2 : greater-than output condition

Table 1-2 Parameter Description

Parameter	Values	Description
width	≥ 4	Word width of a and b
p1_width	2 to *width*–2	Word width of part1 of duplex compare

Table 1-3 Synthesis Implementations[a]

Implementation Name	Function	License Feature Required
rpl	Ripple carry synthesis model	DesignWare
bk	Brent-Kung synthesis model	DesignWare

a. During synthesis, Design Compiler selects the appropriate architecture for your constraints. However, you may force Design Compiler to use one of the architectures described in this table.

Table 1-4 Simulation Models

Model	Function
DW01.DW_CMP_DX_CFG_SIM	Design unit name for VHDL simulation
dw/dw01/src/DW_cmp_dx_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW_cmp_dx.v	Verilog simulation model source code

Block Diagram

The control signal `tc` determines whether the input data is interpreted as unsigned (`tc` is 0) or signed (`tc` is 1) numbers. Figure 1-1 shows a functional block diagram of DW_cmp_dx.

Figure 1-1 Functional Block Diagram

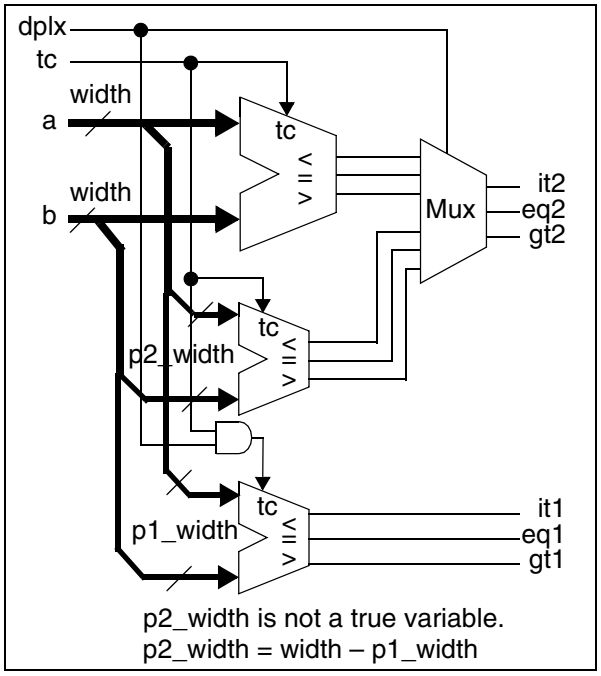


Table 1-5 Operating Modes

dplx	tc	Function
0	0	Simplex unsigned compare operation:[a] ■ lt2 is 1 when a is less than b ■ eq2 is 1 when a is equal to b ■ gt2 is 1 when a is greater than b
0	1	Simplex signed compare operation:[a] ■ lt2 is 1 when a is less than b ■ eq2 is 1 when a is equal to b ■ gt2 is 1 when a is greater than b

Table 1-5 Operating Modes (Continued)

dplx	tc	Function
1	0	Duplex unsigned compare operation: ■ lt2 is 1 when part2 a is less than part2 b ■ eq2 is 1 when part2 a is equal to part2 b ■ gt2 is 1 when part2 a is greater than part2 b ■ lt1 is 1 when part1 a is less than part1 b ■ eq1 is 1 when part1 a is equal to part1 b ■ gt1 is 1 when part1 a is greater than part1 b
1	1	Duplex signed compare operation: ■ lt2 is 1 when part2 a is less than part2 b ■ eq2 is 1 when part2 a is equal to part2 b ■ gt2 is 1 when part2 a is greater than part2 b ■ lt1 is 1 when part1 a is less than part1 b ■ eq1 is 1 when part1 a is equal to part1 b ■ gt1 is 1 when part1 a is greater than part1 b

a. The output values of lt1, eq1, and gt1 in the simplex modes are “don’t care.”

Related Topics

- Math – Arithmetic Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DW_foundation_comp.all;

entity DW_cmp_dx_inst is
  generic ( inst_width    : NATURAL := 24;
            inst_p1_width : NATURAL := 16 );
  port ( inst_a    : in std_logic_vector(inst_width-1 downto 0);
         inst_b    : in std_logic_vector(inst_width-1 downto 0);
         inst_tc   : in std_logic;
         inst_dplx : in std_logic;
         lt1_inst  : out std_logic;
         eq1_inst  : out std_logic;
         gt1_inst  : out std_logic;
         lt2_inst  : out std_logic;
         eq2_inst  : out std_logic;
         gt2_inst  : out std_logic );
end DW_cmp_dx_inst;

architecture inst of DW_cmp_dx_inst is
begin

  -- Instance of DW_cmp_dx
  U1 : DW_cmp_dx
    generic map ( width => inst_width, p1_width => inst_p1_width )
    port map ( a => inst_a, b => inst_b, tc => inst_tc, dplx => inst_dplx,
               lt1 => lt1_inst, eq1 => eq1_inst, gt1 => gt1_inst,
               lt2 => lt2_inst, eq2 => eq2_inst, gt2 => gt2_inst );
end inst;

-- pragma translate_off
configuration DW_cmp_dx_inst_cfg_inst of DW_cmp_dx_inst is
  for inst
  end for; -- inst
end DW_cmp_dx_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW_cmp_dx_inst( inst_a, inst_b, inst_tc, inst_dplx, lt1_inst,
                       eq1_inst, gt1_inst, lt2_inst, eq2_inst, gt2_inst );

  parameter width = 24;
  parameter p1_width = 16;

  input [width-1 : 0] inst_a;
  input [width-1 : 0] inst_b;
  input inst_tc;
  input inst_dplx;
  output lt1_inst;
  output eq1_inst;
  output gt1_inst;
  output lt2_inst;
  output eq2_inst;
  output gt2_inst;

  // Instance of DW_cmp_dx
  DW_cmp_dx #(width, p1_width)
    U1 ( .a(inst_a), .b(inst_b), .tc(inst_tc), .dplx(inst_dplx),
         .lt1(lt1_inst), .eq1(eq1_inst), .gt1(gt1_inst),
         .lt2(lt2_inst), .eq2(eq2_inst), .gt2(gt2_inst) );
endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 5 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com